An algorithm for the linear complementarity problem with upper and lower bounds

Talman, A.J.J.; van der Laan, G.

Published in:
Journal of Optimization Theory and Applications

Publication date:
1989

JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 62, No. 1, JULY 1989

An Algorithm for the Linear Complementarity Problem with Upper and Lower Bounds[1]

G. VAN DER LAAN[2] AND A. J. J. TALMAN[3]

Communicated by F. Zirilli

Abstract. In this paper, we adapt the octahedral simplicial algorithm for solving systems of nonlinear equations to solve the linear complementarity problem with upper and lower bounds. The proposed algorithm generates a piecewise linear path from an arbitrarily chosen point z^0 to a solution point. This path is followed by linear programming pivot steps in a system of n linear equations, where n is the size of the problem. The starting point z^0 is left in the direction of one of the 2^n vertices of the feasible region. The ray along which z^0 is left depends on the sign pattern of the function value at z^0. The sign pattern of the linear function and the location of the points in comparison with z^0 completely govern the path of the algorithm.

Key Words. Linear complementarity problem, pivoting algorithm, arbitrary starting point.

1. Introduction

The linear complementarity problem (LCP) consists in finding two vectors s and z in R^n such that, for a given $n \times n$ matrix M and n-vector q,

(i) $s = Mz + q$,
(ii) $s, z \geq 0$,
(iii) $s^{\mathrm{T}} z = 0$.

The LCP is an important problem in mathematical programming [see, e.g., Garcia and Gould (Ref. 1)]. Lemke (Ref. 2) first presented a solution

[1] This research is part of the VF-Program "Equilibrium and Disequilibrium in Demand and Supply," approved by the Netherlands Ministry of Education, Den Haag, The Netherlands.

[2] Professor, Department of Economics and Econometrics, Free University, Amsterdam, The Netherlands.

[3] Professor, Department of Econometrics, Tilburg University, Tilburg, The Netherlands.

for this problem. Lemke's algorithm is initialized at $z=0$ and traces by pivoting from this point a piecewise linear path of points until a solution is obtained or a ray is encountered. Talman and Van der Heyden (Ref. 3) proposed a class of pivoting algorithms which allow for an arbitrary starting point in the nonnegative orthant. An element of this class is characterized by a number between $n+1$ and $2n$, being the number of different directions in which a starting point z^0 in the interior of the orthant can be left. Along which direction the starting point is left depends on the value of the components of s at z^0. When $z^0=0$ is chosen to be the starting point, these algorithms reduce to Lemke's algorithm. However, in such applications as parametric studies or solving a nonlinear complementarity problem through a sequence of approximating LCP's (see, e.g., Ref. 4 or Ref. 5), the feature of allowing arbitrarily chosen starting points has obvious practical merit.

In this paper, we consider an LCP with lower and upper bounds. This problem consists of finding vectors s and z in R^n such that, for a given matrix M, an n-dimensional vector q, and two n-dimensional vectors a and b with $a_j < b_j$, $j=1,\ldots,n$,

$$s = Mz + q, \tag{1a}$$

$$a \leq z \leq b, \tag{1b}$$

$$z_j = a_j \rightarrow s_j \leq 0, \tag{1c}$$

$$a_j < z_j < b_j \rightarrow s_j = 0, \qquad j=1,\ldots,n, \tag{1d}$$

$$z_j = b_j \rightarrow s_j \geq 0. \tag{1e}$$

The LCP with lower and upper bounds can easily be converted to a classical LCP on R^{2n}, so that the algorithms of Lemke and of Talman and Van der Heyden could be applied for solving the problem. However, by Everts (Ref. 6), the algorithm of Talman and Van der Heyden with $2n$ directions was generalized in order to solve the LCP with upper and lower bounds directly. When the paths of points generated by Everts' algorithm hits a boundary face of the feasible region,

$$C^n = \{z \in R^n \mid a \leq z \leq b\},$$

the algorithm continues in this face. To reach a solution point in a k-face of C^n, $0 \leq k \leq n$, from an interior point of C^n, the algorithm needs at least $2n-k$ pivot steps if $k \geq 1$ and $2n-1$ steps if $k=0$, i.e., if one of the vertices of C^n is found as a solution point. We notice that the algorithm is an adaption of the cubical algorithm for solving a system of nonlinear equations presented in Ref. 7; see also Ref. 8.

In this paper, we propose a pivoting algorithm to solve (1) having 2^n directions to leave the arbitrarily chosen starting point z^0 in C^n. This algorithm is an adaption of the octahedral algorithm for solving a system of nonlinear equations [see Wright (Ref. 9)]. From z^0, there points a ray to each vertex of C^n. The ray along which the algorithm leaves z^0 depends on the sign pattern of $s^0 = Mz^0 + q$. The piecewise linear path of points z traced by the algorithm is completely determined by the sign pattern of $s = Mz + q$ and the location of z with respect to the starting point. The algorithm is such that a solution point is found as soon as the path of the algorithm hits a face of C^n not containing the starting point. This motivates the presentation of the new algorithm. Because a solution is found as soon as the boundary of C^n is reached, we have that, if along the initial ray the sign pattern of s does not change, the end point of this ray solves problem (1), and therefore the solution will be found in just one pivot step. Note that such a solution is a vertex of C^n. In general, for some k, $0 \le k \le n-1$, a solution on a k-face of C^n could be found in $k+1$ pivot steps, which is considerably less than for the algorithm with $2n$ rays when k is small with respect to n.

The paper is organized as follows. In Section 2, we present a detailed description of the piecewise linear path followed by the algorithm. The steps of the algorithm are presented in Section 3. Finally, in Section 4, we consider the case that some or all of the upper bounds are infinite.

2. Movements of the Algorithm

We consider points (s, z) in $R^n \times C^n$ satisfying

$$s = Mz + q, \tag{2}$$

where

$$C^n = \{z \in R^n \mid a_i \le z_i \le b_i,\ i = 1, \ldots, n\},$$

with $-\infty < a_i < b_i < \infty$ for all i. The case that some of the numbers b_i (a_i) are infinite (minus infinite) is discussed in Section 5. Starting at an arbitrary point z^0, the algorithm adjusts z by increasing z_i if $s_i > 0$ and $z_i^0 < b_i$ and by decreasing z_i if $s_i < 0$ and $z_i^0 > a_i$. More precisely, for each sign vector t in $\{-1, +1\}^n$, a direction $d(t) = v(t) - z^0$ is defined with $v(t)$, the vertex of C^n, given by

$$v_i(t) = b_i, \qquad \text{if } t_i = 1; \qquad v_i(t) = a_i, \qquad \text{if } t_i = -1.$$

Assuming that $s_i^0 \ne 0$, for all i, an initial movement is made from z^0 in the direction $d(t^0)$ toward the vertex $v(t^0)$ of C^n with $t^0 = \operatorname{sign} s^0$. So, the direction in which z^0 is left is the direction associated with $\operatorname{sign} s^0$. If $t = \operatorname{sign}(Mz + q)$ does not change when z moves along the ray from z^0 to

$v(t^0)$, we have that $v(t^0)$ is a solution point and this point is found after just one pivot step. Otherwise, the ray to $v(t^0)$ is left as soon as, for some i, s_i becomes equal to zero, say at the point z^1. So, for each point z between z^0 and z^1, we have that $t = \text{sign}(Mz + q) = t^0$. So, for each z on the ray from z^0 to $v(t^0)$ between z^0 and z^1, the complementarity property between z and t holds; that is, for some λ, $0 \leq \lambda \leq 1$,

$$z_j = z_j^0 + \lambda(b_j - z_j^0), \qquad \text{for all } j \text{ with } t_j = +1,$$
$$z_j = z_j^0 + \lambda(a_j - z_j^0), \qquad \text{for all } j \text{ with } t_j = -1.$$

At the point z^1 where s_i is equal to zero, the algorithm continues by making a pivot step in which z_i is decreased away from

$$z_i^1 = z_i^0 + \lambda(b_i - z_i^0),$$

if t_i changes from $+1$ to 0, and in which z_i is increased away from

$$z_i^1 = z_i^0 + \lambda(a_i - z_i^0),$$

if t_i changes from -1 to 0, while s_i is kept equal to zero. In general, as shown below, a piecewise linear path of points is generated such that for each z on the path the complementarity property between z and $t = \text{sign } s = \text{sign}(Mz + q)$ given by

$$z_j = z_j^0 + \lambda(b_j - z_j^0), \qquad \text{for all } j \text{ with } t_j = +1, \tag{3a}$$
$$z_j = z_j^0 + \lambda(a_j - z_j^0), \qquad \text{for all } j \text{ with } t_j = -1, \tag{3b}$$
$$z_j^0 + \lambda(a_j - z_j^0) \leq z_j \leq z_j^0 + \lambda(b_j - z_j^0), \qquad \text{for all } j \text{ with } t_j = 0 \tag{3c}$$

holds for some λ, $0 < \lambda \leq 1$. This complementarity property governs the algorithm and connects z^0 with a solution to problem (1) through the piecewise linear path of points. A solution point is reached as soon as either $t_j = 0$, for all j with $z_j \neq z_j^0$, or $\lambda = 1$. In the first case, we have that $s_j = 0$, if $z_j \neq z_j^0$, while (3) implies that $z_j = z_j^0$, iff either $z_j^0 = b_j$ and $t_j = +1$ or $z_j^0 = a_j$ and $t_j = -1$. If $\lambda = 1$, then $z_j = b_j$, if $t_j = +1$; $z_j = a_j$, if $t_j = -1$; and $t_j = 0$, if $a_j < z_j < b_j$.

We now prove that the points (s, z) in $R^n \times C^n$ satisfying (3) with $t = \text{sign } s$ indeed induce a sequence of adjacent line segments in C^n connecting z^0 with a solution point. Let T be the set of nonzero sign vectors in R^n, i.e.

$$T = \{\in R^n \mid t_i \in \{-1, 0, +1\},\ i = 1, \ldots, n,$$
$$\text{and } t_i \neq 0 \text{ for at least one } i\}.$$

The point z^0, an arbitrary point in C^n, is the starting point of the algorithm. Then, for $t \in T$, we define the convex polyhedral set $A(t)$ by

$$A(t) = \varnothing, \qquad \text{if } z_i^0 = a_i \text{ for all } i \text{ with } t_i = -1,$$
$$\text{and } z_i^0 = b_i \text{ for all } i \text{ with } t_i = +1,$$

and otherwise

$$\begin{aligned} A(t)=\{z\in C^n \mid z_j &= \lambda(b_j-z_j^0)+z_j^0, \text{ if } t_j=+1,\\ z_j &= \lambda(a_j-z_j^0)+z_j^0, \text{ if } t_j=-1,\\ \lambda(a_j-z_j^0)\le z_j-z_j^0 &\le \lambda(b_j-z_j^0), \text{ if } t_j=0,\\ &\text{with } 0\le\lambda\le 1\}. \end{aligned} \tag{4}$$

For $t\in T$, let

$$I^-(t)=\{i\in I_n \mid t_i=-1\}, \qquad I(t)=\{i\in I_n \mid t_i=0\},$$
$$I^+(t)=\{i\in I_n \mid t_i=+1\},$$

where $I_n=\{1,\ldots,n\}$. Then, the dimension of a nonempty set $A(T)$ is equal to

$$\dim A(t)=|I(t)|+1.$$

If $t\in\{-1,+1\}^n$, then $\dim A(t)=1$ [unless $z^0=v(t)$] and $A(t)$ is the line segment connecting z^0 with $v(t)$, i.e., $A(t)$ is the ray along which the direction $d(t)=v(t)-z^0$ points at z^0. So, if z^0 is not a vertex of C^n, there are 2^n directions along one of which z^0 is left. When z^0 is a vertex, then there are 2^n-1 directions. The algorithm leaves z^0 along the ray $A(t^0)$. According to (3), the algorithm generates points z in C^n such that, for some t in T, both z lies in $A(t)$ and $t=\operatorname{sign} s$. For $t\in T$, let $C(t)$ be defined by

$$C(t)=\mathrm{Cl}\{z\in C^n \mid \operatorname{sign}(Mz+q)=t\},$$

and let

$$B(t)=A(t)\cap C(t).$$

A point z satisfies (3) if and only if z lies in $B(t)$ for some $t\in T$, i.e., if z lies in A(sign s). We now introduce basic and nonbasic variables.

Definition 2.1. For some $z\in C^n$, $z\neq z^0$, let $A(t')$ be the smallest set $A(t)$ containing z in its relative interior. Then, the variable z_j, $j\in I_n$, is nonbasic if $t'_j\neq 0$. With $s=Mz+q$, the variable s_j is nonbasic if $s_j=0$. Furthermore, let λ be defined as in (3) with $t=t'$. Then, λ is nonbasic if $\lambda=1$. Finally, for $z=z^0$, λ is defined to be equal to zero and all variables z_j, $j\in I_n$, and λ are nonbasic. When not nonbasic, a variable is basic.

We call a pair (s, z) complementary if, for each $j\in I_n$, either or both z_j and s_j are nonbasic. Notice that (s, z) is complementary if z lies in $B(t)$ for some $t\in T$.

Assumption 2.1. For each z in $B(t)$, $t \in T$, it holds that, among the $2n+1$ variables (z, s, λ) with $s = Mz + q$ and λ as defined in (3) ($\lambda = 0$ if $z = z^0$), at most $n+1$ variables are nonbasic.

This nondegeneracy assumption does not cause a loss of generality, since if degeneracy occurs the assumption will be restored by a slight perturbation of the data (M, q); i.e., the assumption holds for almost all pairs of data (M, q).

We now show that, for each t in T, the set $B(t)$ is either empty or a line segment with two endpoints and that, if an endpoint of a nonempty $B(t)$ is neither the starting point z^0 nor a solution point, it is an endpoint of a line segment $B(t')$, with t' differing from t in just one component. By definition, the pair (s^0, z^0) is complementary because at z^0 all the variables z_j are nonbasic. Since λ is also nonbasic at z^0, Assumption 2.1 implies that $s_j^0 \neq 0$ for all j. Hence, z^0 is an endpoint of $B(\text{sign } s^0)$ and lies in no other set $B(t)$, $t \neq t^0 = \text{sign } s^0$. By increasing λ away from zero, a line segment of complementary points (s, z) in $B(t^0)$ is obtained. At a certain point z (one and just one, because of Assumption 2.1) a basic variable becomes nonbasic. At this endpoint z, either $\lambda = 1$ or $s_j = 0$ for just one $j \in I_n$. In the first case, a solution has been reached, as has been shown above. In the latter case, the point z also lies in $B(t)$, with $t = \text{sign } s$, so that z_j becomes basic and $t_j = 0$. In general, under Assumption 2.1, each nonempty $B(t)$ is a line segment of points z complementary to s in $A(t)$ having two endpoints. An endpoint is characterized by the fact that $n+1$ variables are nonbasic. Hence, at an endpoint z of $B(t)$, either λ is equal to 0 or 1 and, for all j, either z_j or s_j is nonbasic or λ is basic and, for exactly one index h, both z_h and s_h are nonbasic. If λ is equal to 0, then $z = z^0$.

In the following lemmas, we consider the endpoints of line segments in case λ is not equal to 0.

Lemma 2.1. Let z be an endpoint of a line segment $B(t)$. If λ is equal to 1, then z is a solution point.

Proof. Since $z \in A(t)$ and $\lambda = 1$, we have that

$$\begin{aligned} z_j &= b_j, && \text{if } t_j = +1, \\ z_j &= a_j, && \text{if } t_j = -1, \\ a_j \leq z_j &\leq b_j, && \text{if } t_j = 0. \end{aligned}$$

Moreover, $t = \text{sign}(Mz + q)$, since z also lies in $C(t)$ and because of Assumption 2.1. Hence, z is a solution point. □

In the next lemma, let $Z^b(Z^a)$ be the set of indices j for which $z_j^0 = b_j(z_j^0 = a_j)$.

Lemma 2.2. Let z be an endpoint of a line segment $B(t)$, and let $s = Mz + q$. When $s_h = 0$ for some h with $t_h \neq 0$, then z is a solution point if

$$I^+(t)\backslash\{h\} \subset Z^b \qquad \text{and} \qquad I^-(t)\backslash\{h\} \subset Z^a.$$

Proof. The conditions of the lemma imply that $z_j^0 = b_j$ for all $j \in I^+(t)$ and $z_j^0 = a_j$ for all $j \in I^-(t)$, $j \neq h$. Furthermore, $t_j = \operatorname{sign} s_j$, for all $j \neq h$ and $s_h = 0$. Therefore, for $j \neq h$, $j \in I^+(t)$ implies

$$s_j > 0 \qquad \text{and} \qquad z_j = \lambda(b_j - z_j^0) + z_j^0 = b_j,$$

and $j \in I^-(t)$ implies

$$s_j < 0 \qquad \text{and} \qquad z_j = \lambda(a_j - z_j^0) + z_j^0 = a_j.$$

Moreover, for all other indices j, we have that

$$s_j = 0 \qquad \text{and} \qquad a_j \leq z_j \leq b_j.$$ □

Lemma 2.3. Let z be an endpoint of a line segment $B(t)$ and let $s = Mz + q$. If $s_h = 0$ for some h with $t_h \neq 0$ and $I^+(t)\backslash\{h\}$ contains at least one index j, $j \neq h$, not in Z^b or $I^-(t)\backslash\{h\}$ contains at least one index j, $j \neq h$, not in Z^a, then z is also an endpoint of $B(t')$ with $t'_h = 0$ and $t'_j = t_j$ for all $j \neq h$.

Proof. The conditions of the lemma imply that there is an index j, $j \neq h$, with $t'_j = +1$ and $z_j^0 < b_j$ or an index k, $k \neq h$, with $t'_k = -1$ and $z_k^0 > a_k$. Hence, $A(t')$ is nonempty. Moreover, since $t'_j = t_j$, $j \neq h$, $t'_h = 0$, and $t_h \in \{-1, +1\}$, $A(t')$ contains $A(t)$ as a boundary facet, and therefore z is in the boundary of $A(t')$. Finally, since $t_j = \operatorname{sign} s_j$, $j \neq h$, and $s_h = 0$, we have that $t' = \operatorname{sign} s$. Consequently, z lies in $C(t')$ and is an endpoint of $B(t')$. □

Lemma 2.4. Let z be an endpoint of a line segment $B(t)$, and let $s = Mz + q$. If z_h is nonbasic for some h with $t_h = 0$, then z is also an endpoint of $B(t')$ with $t'_j = t_j$, $j \neq h$, and $t'_h \neq 0$.

Proof. At z, the variable z_h becomes nonbasic, i.e., for the λ defined in (3), it holds that

$$\lambda(a_h - z_h^0) \leq z_h - z_h^0 \leq \lambda(b_h - z_h^0),$$

with one of the inequalities an equality. If

$$z_h = z_h^0 + \lambda(a_h - z_h^0),$$

then $z \in A(t')$ with $t'_h = -1$ and $t'_j = t_j$, $j \neq h$. On the other hand, if

$$z_h = z^0_h + \lambda(b_h - z^0_h),$$

then $z \in A(t')$ with $t'_h = +1$ and $t'_j = t_j$, $j \neq h$. Finally, since $t = \operatorname{sign} s$ with $s_h = 0$, we have that z lies in the boundary of $C(t')$. Hence, z is an endpoint of $B(t')$. □

Lemmas 2.3 and 2.4 say that, if z' is an endpoint of the line segment $B(t)$ and if z' is neither a solution point nor z^0, then z' is an endpoint of exactly one other line segment $B(t')$. The nondegeneracy assumption guarantees that, at z', just one basic variable becomes nonbasic. This implies that t' is uniquely determined. So, linking the line segments $B(t)$ for various $t \in T$ together, the set $B = \bigcup_{t \in T} B(t)$ contains a piecewise linear path having the starting point z^0 as an endpoint. Since T consists of a finite number of sign vectors t, and since each $B(t)$ is either empty of just a single line segment, the path in B originating at z^0 consists of a finite number of linear pieces and must end with a solution point z'. This path is generated by the algorithm and can be followed by a sequence of linear programming steps in a system of n linear equations. In general, the set B consists of p.l. loops and paths with two endpoints. Exactly one path connects z^0 with a solution point and all other paths connect two solution points. The total number of solutions to the LCP with upper and lower bounds is therefore (in general) odd.

3. Steps of the Algorithm

Let us consider now a point z on the path traced by the algorithm, i.e., $z \in A(t) \cap C(t)$, for some sign vector $t \in T$. The fact that $z \in A(t)$ implies that

$$\begin{aligned}
&z_j = \lambda(b_j - z^0_j) + z^0_j, && \text{if } t_j = +1,\\
&z_j = \lambda(a_j - z^0_j) + z^0_j, && \text{if } t_j = -1,\\
&\lambda(a_j - z^0_j) \le z_j - z^0_j \le \lambda(b_j - z^0_j), && \text{if } t_j = 0,
\end{aligned}$$

with $0 \le \lambda \le 1$. Let $t' \in \{-1, +1\}^n$ be an arbitrary sign vector such that $t'_i = t_i$ if $t_i \neq 0$. Then, we can rewrite z as

$$z = (1 + \lambda)z^0 + \lambda v(t') - \Sigma_{h \in I(t)}\, \delta_h t'_h e(h), \tag{5}$$

for certain δ_h, $0 \le \delta_h \le \lambda(b_h - a_h)$, where $e(h)$ is the hth unit vector in R^n, $h = 1, \ldots, n$. From (5), we obtain

$$s = Mz + q = (1 - \lambda)Mz^0 + \lambda Mv(t') - \Sigma_{h \in I(t)}\, \delta_h t'_h Me(h) + q. \tag{6}$$

Setting

$$Mz^0 + q \equiv q^0,$$

(6) becomes

$$s+\lambda M[z^0-v(t')]+\Sigma_{h\in I(t)}\delta_h t'_h M_h=q^0, \tag{7}$$

where $M_h=Me(h)$ is the hth column of M. On the other hand, $z\in C(t)$ implies that

$$s=\Sigma_{h\notin I(t)}\mu_h t_h e(h), \qquad \text{with } \mu_h=t_h s_h\geq 0.$$

All of this together gives that z lies in $B(t)$ if and only if the equation

$$\lambda M[z^0-v(t')]+\Sigma_{h\in I(t)}\delta_h t'_h M_h+\Sigma_{h\notin I(t)}\mu_h t_h e(h)=q^0 \tag{8}$$

holds with $0\leq\lambda\leq 1$, $0\leq\delta_h\leq\lambda(b_h-a_h)$, $h\in I(t)$, $\mu_h\geq 0$, $h\notin I(t)$. The sign vector t' containing no zeros will be automatically generated by the algorithm. The nondegeneracy Assumption 2.1 implies that, at any solution to the system (8) of n linear equations, at most one of the $n+1$ variables λ, δ_h, $h\in I(t)$, μ_h, $h\notin I(t)$, is equal to an (upper or lower) bound. Hence, a linear path of points $B(t)$, $t\in T$, can be followed by making a linear programming (l.p.) pivot step in the system (8). Notice that the system (8) consists of $n+1$ columns. For each h, $h=1,\ldots,n$, there is the column M_h (with sign t'_h) if $h\in I(t)$ and the unit vector column $e(h)$ (with sign t_h) if $h\notin I(t)$. Moreover, there is an additional column $M[z^0-v(t')]$. Recall that $v(t')$ is a vertex of C^n. If, at a solution to (8), one of the variables λ, δ_h, or μ_h is equal to a bound, then the corresponding z defined in (5) is an endpoint of the line segment $B(t)$. In particular, $\mu_h=0$ corresponds to $s_h=0$, $h\notin I(t)$. The piecewise linear path of points in B from z^0 to a solution point can therefore be followed by making a sequence of linear programming pivot steps in the system (8) with either a column of the matrix M or a unit vector.

Step 0. Initialization. Set

$$t'=t=t^0=\text{sign}(Mz^0+q).$$

if $z^0=v(t^0)$, then (s^0, z^0) solves the problem. Otherwise, make an l.p. pivot step, with $M[z^0-v(t')]$ in the system (8).

Step 1. If $\lambda=1$, then a solution is found (see Lemma 2.1). If $\delta_i=0$ for some $i\in I(t)$, go to Step 3. If $\delta_i=\lambda(b_i-a_i)$ for some $i\in I(t)$, set t'_i equal to $-t'_i$, adapt the column $M[z^0-v(t')]$ in (8) accordingly, and go to Step 3. If $\mu_i=0$ for some $i\notin I(t)$ and if, for all other $h\notin I(t)$,

$$z_h^0=b_h, \qquad \text{if } t_h=+1, \qquad z_h^0=a_h, \qquad \text{if } t_h=-1,$$

then a solution is found (see Lemma 2.2).

Step 2. See Lemma 2.3. Set $t_i=0$, make an l.p. pivot step with t'_iM_i in the system (8); go to Step 1.

Step 3. See Lemma 2.4. Set $t_i=t'_i$, make an l.p. pivot step with $t_ie(i)$ in the system (8); go to Step 1.

The algorithm is illustrated for $n=2$ in Fig. 1. In this figure, there are three solution points, namely, x^*, x^1, and v. The algorithm traces the path from z^0, along a, b, c, and d to v, which takes 5 l.p. steps. At z^0, we have that $s_1^0>0$ and $s_2^0<0$, implying a search in the direction $d[+1,-1)^\top]$. Then $s_2=0$ is followed from a until the ray $A[(+1,+1)]^\top]$ is hit at b. This ray is followed till c where $s_1=0$. Then, the line $s_1=0$ is followed in $A[(0,+1)^\top]$ until the point d in $A[(-1,+1)^\top]$. This ray is followed until the solution point v is found. We notice that the maximum number of steps the algorithm can take is 5 in the case $n=2$, so that this example shows the worst case. In comparison, the maximum number of steps is equal to 11 when the algorithm with $2n$ rays [see Everts (Ref. 6)] is taken.

Finally, we notice that each point z satisfying (3) for some λ solves the LCP with a vector of lower bounds,

$$a(\lambda)=z^0+\lambda(a-z^0),$$

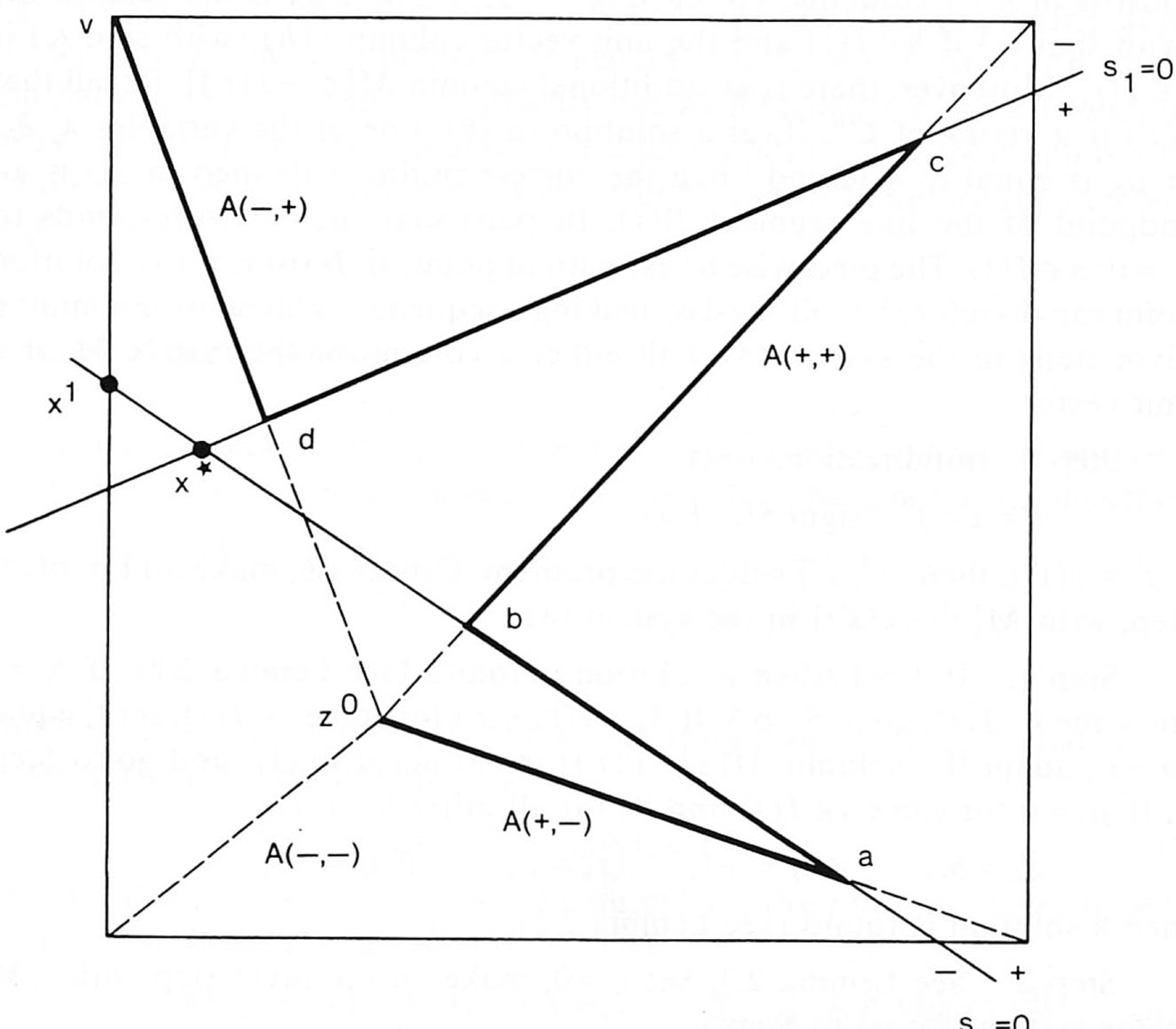

Fig. 1. The worst case for the octahedral algorithm, $n=2$; $A(+,-)$ denotes $A(t)$ with $t=(+,-)$, etc.

and a vector of upper bounds,

$$b(\lambda)=z^0+\lambda(b-z^0).$$

So, for varying λ, $0\leq\lambda\leq 1$, the algorithm generates from z^0 with $\lambda=0$ points which solve the problem (1) on the cube

$$C^n(\lambda)=\{z\in R^n\,|\,\lambda a+(1-\lambda)z^0\leq z\leq \lambda b+(1-\lambda)z^0\}.$$

4. Unbounded Case

The algorithm described in this paper can easily be adapted for the case that some of the a_i's or b_i's are not finite. We may assume that, for each j, either b_j or a_j is finite or both. For simplicity, let a_j be finite for all j, and let J be the set of indices for which b_j is infinite. To modify the subdivision of the region

$$C=\{z\in R^n\,|\,a_j\leq z_j\leq b_j, j\notin J;\ a_j\leq z_j, j\in J\}$$

into regions $A(t)$, we restrict the set of admissible sign vectors to $T^1\cup T^2$ with

$$T^1=\{t\in T\,|\,t_j=0,\text{ if } j\notin J,\text{ and } t_j\geq 0,\text{ if } j\in J\},$$

$$T^2=\{t\in T\,|\,t_j\leq 0 \text{ if } j\in J\}.$$

Notice, that, for each admissible t, the components in J are either all nonpositive or all nonnegative. Moreover, $T^2=T$ if $J=\varnothing$, in which case we define T^1 to be empty. For $t\in T^2$, we define the sets $A(t)$ and $C(t)$ as in Section 2. Notice, however, that if for some $j\in J$, $t_j=0$, we obtain

$$z_j\geq z_j^0+\lambda(a_j-z_j^0),$$

and hence $\lambda=0$ does not imply $z=z^0$. For $t\in T^1$, we redefine $A(t)$ by

$$A(t)=\{z\in C\,|\,z_j=z_j^0,\text{ if } t_j=0,\text{ and } z_j\geq z_j^0,\text{ if } t_j=+1\}.$$

So, the dimension of $A(t)$, $t\in T^1$, is equal to the number of nonzeros of the vector t. In addition, with $s=Mz+q$, we redefine $C(t)$, $t\in T^1$, by

$$C(t)=\{z\in C\,|\,\beta\geq 0 \text{ and } s_j=\beta,\text{ if } t_j=+1,\text{ with } \beta=\max_{h\in J}s_h\}.$$

So, the dimension of $C(t)$, $t\in T^1$, is equal to $n-|I^+(t)|+1$. Observe that T^1 only allows nonnegative sign vectors with zero components for all j not in J. If for $t\in T^1$ the algorithm moves into a region $A(t)$, we require that the components s_j with $t_j=+1$ are equal to each other and nonnegative and that they are greater than the other components in J.

Under the nondegeneracy assumption, we now obtain that, for each $t \in T^1 \cup T^2$, $A(t) \cap C(t)$ is either empty or a line segment with one or two endpoints. A line segment with one endpoint is a half-ray going to infinity in at least one component. Again, each endpoint of a line segment $B(t) = A(t) \cap C(t)$ is either the starting point z^0, or a solution to problem (1), or an endpoint of just one other line segment different from $B(t)$. Let $s^0 = Mz^0 + q$, and suppose that $\max_{h \in J} s_h^0 > 0$. Then, z^0 is an endpoint of the line segment $B(t^0)$ with $t_j^0 = +1$ for the unique index $j \in J$ for which $s_j^0 = \max_{h \in J} s_h^0$, and $t_h^0 = 0$ for all $h \neq j$. If $\max_{h \in J} s_h^0 < 0$ or J is empty, then z^0 is an endpoint of the line segment $B(t^0)$ with $t^0 = \text{sign } s^0$ in T^2. If, however, in the latter case $z^0 = v(t^0)$, then z^0 solves problem (1). Now, for some $t \in T^1$, let z, $z \neq z^0$, be an endpoint of $B(t)$, and let $s = Mz + q$. At such an endpoint, either $\max_{h \in J} s_h = 0$, or $s_h = \max_{k \in J} s_k$, for some $h \in J$ with $t_h = 0$, or $z_h = z_h^0$, for some $h \in J$ with $t_h = +1$. First, suppose that $\max_{h \in J} s_h = 0$. Then, z is a solution of problem (1) if, for all k with $t_k = 0$, $z_k^0 = b_k$, when $s_k > 0$, and $z_k^0 = a_k$, when $s_k < 0$. Otherwise, z is an endpoint of the line segment $B(t')$ with t' in T^2 equal to $\text{sign}(Mz + q)$. Second, if $s_h = \max_{k \in J} s_k$ for some $h \in J$ with $t_h = 0$, then z is also an endpoint of $B(t')$ with $t_h' = +1$ and $t_j' = t_j$ for all $j \neq h$. Finally, if $z_h = z_h^0$ for some $h \in J$ with $t_h = +1$, then z is also an endpoint of $B(t')$ with $t_h' = 0$ and $t_j' = t_j$ for all $j \neq h$.

In case $t \in T^2$, an endpoint z, $z \neq z^0$, of $B(t)$ is either a solution point or an endpoint of exactly one other line segment $B(t')$ with t' differing from t in just one component as described in Section 2, or at z the variable λ is 0. In this case, we have $z_j = z_j^0$ if $j \notin J$, while, for $j \in J$, $z_j = z_j^0$ if $t_j = -1$ and $z_j \geq z_j^0$ if $t_j = 0$. Since $z \neq z^0$, we must have $z_j > z_j^0$ for at least one $j \in J$, so that z is also an endpoint of $B(t')$ with t' in T^1 such that $t_j' = 0$ for $j \notin J$, while, for $j \in J$, $t_j' = +1$ if $t_j = 0$ and $t_j' = 0$ if $t_j = -1$.

By linking the nonempty sets $B(t)$ in this way together, we obtain a collection of piecewise linear paths and loops in C. One path originates at z^0, and this path can be followed by making linear programming pivot steps in a system of linear equations. If $t \in T^1$, the l.p. pivot steps are made in the system

$$\Sigma_{h \in K} \delta_h M_h + \Sigma_{h \in J/K} \mu_h e(h) + \Sigma_{h \notin J} \vartheta_h e(h) - \beta e = -q^0, \tag{9}$$

with

$$K = \{k \in J \mid t_k = 0\},$$

with e the n-vector with all components equal to one, and with feasible solutions $\delta_h \geq 0$, $\mu_h \geq 0$, $\vartheta_h \in R$, and $\beta \geq 0$. This system is similar to the system of equations in which l.p. pivot steps are made for Lemke's algorithm. Observe that the pivot steps are such that, for each $h \in J$, δ_h and μ_h are always complementary. If $t \in T^2$, the l.p. pivot steps are made in the system

(8), keeping δ_h and μ_h complementary for all h. If, for some $t \in T^1$, through a pivot step in the system (9), β becomes equal to zero, then the algorithm continues with a pivot step in (8) with $t' = \text{sign } s$ by increasing λ from zero. Conversely, if for some $t \in T^2$, a pivot step in (8) makes λ equal to zero, then the algorithm switches to (9) by increasing β from zero. The path followed in this way terminates with either a solution to problem (1) or a half-ray. Convergence conditions in order to guarantee that the algorithm will not terminate with a half-ray can easily be obtained from such conditions for the classical LCP on R^n_+; see, e.g., Jones (Ref. 10). We remark that, if $J = \{1, \ldots, n\}$, the algorithm proposed above is very similar to the algorithm with $n+1$ rays and arbitrary starting point developed in Ref. 3.

References

1. GARCIA, G. B., and GOULD, F. J., *Studies in Linear Complementarity*, University of Chicago, Chicago, Illinois, 1980.
2. LEMKE, C. E., *Bimatrix Equilibrium Points and Mathematical Programming*, Management Science, Vol. 11, pp. 681-689, 1965.
3. TALMAN, A. J. J., and VAN DER HEYDEN, L., *Algorithms for the Linear Complementarity Problem which Allow an Arbitrary Starting Point*, Homotopy Methods and Global Convergence, Edited by B. C. Eaves et al., Plenum Press, New York, New York, pp. 267-286, 1983.
4. JOSEPHY, N., *Newton's Method for Generalized Equations*, University of Wisconsin, Technical Summary Report No. 1965, 1979.
5. MATHIESEN, L., and HANSEN, T., *An Equilibrium Model for an Open Economy with Institutional Constraints on Factor Prices*, Numerical Solution of Highly Nonlinear Problems, Edited by W. Forster, North-Holland, Amsterdam, The Netherlands, pp. 337-360, 1980.
6. EVERTS, I. D., *Het Lineaire Complementariteitsprobleem*, Landbouw-Economisch Instituut, Den Haag, The Netherlands, 1982 (in Dutch).
7. LAAN, G. VAN DER, and TALMAN, A. J. J., *A Class of Simplicial Restart Fixed-Point Algorithms without an Extra Dimension*, Mathematical Programming, Vol. 20, pp. 33-48, 1981.
8. REISER, P. M., *A Modified Integer Labelling for Complementarity Algorithms*, Mathematics of Operations Research, Vol. 6, pp. 129-139, 1981.
8. WRIGHT, A. H., *The Octahedral Algorithm: A New Simplicial Fixed-Point Algorithm*, Mathematical Programming, Vol. 21, pp. 47-69, 1981.
10. JONES, P. C., *Even More with the Lemke Complementarity Algorithm*, Mathematical Programming, Vol. 34, pp. 239-242, 1986.